

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-030515

(43)Date of publication of application : 02.02.1996

(51)Int.Cl.

G06F 12/16  
G06F 13/12  
G06F 15/78  
G11C 16/06

(21)Application number : 06-168165

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 20.07.1994

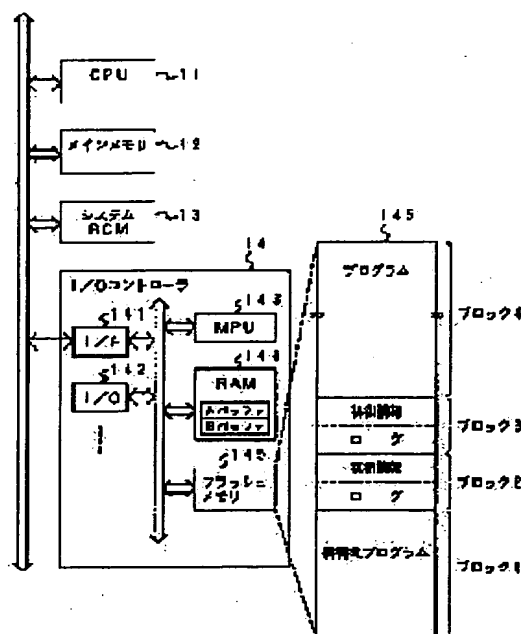
(72)Inventor : KOIZUMI YOSHIYUKI

## (54) DATA PROCESSOR USING FLASH EEPROM

(57)Abstract:

**PURPOSE:** To reduce the number of parts by attaining conventional ROM and EEPROM functions only by a flash EEPROM.

**CONSTITUTION:** The flash EEPROM 145 is divided into four blocks 1 to 4 and a rewriting program, control information and a system program are stored in respective blocks 1 to 4. The control information is duplexedly stored in both the blocks 2, 3. Thereby even when power supply interruption or any other error is caused during the updating of one erasing block, the control information of the other erasing block is normally held. Since control information in a valid information is read out, normal control information can be utilized. Thus the control information can be prevented from being destructed and conventional ROM and EEPROM functions can be attained only by the flash EEPROM 145.



## LEGAL STATUS

[Date of request for examination]

07.06.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3268130

[Date of registration]

11.01.2002

[Number of appeal against examiner's decision of rejection]

BEST AVAILABLE COPY

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-30515

(43) 公開日 平成8年(1996)2月2日

(51) Int.Cl. <sup>6</sup>	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 12/16	3 1 0 J	7623-5B		
13/12	3 4 0 A	7368-5E		
15/78	5 1 0 A			
G 1 1 C 16/06				

G 1 1 C 17/ 00 3 0 9 F  
 審査請求 未請求 請求項の数7 O L (全 14 頁)

(21) 出願番号 特願平6-168165

(22) 出願日 平成6年(1994)7月20日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 小泉 義行

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

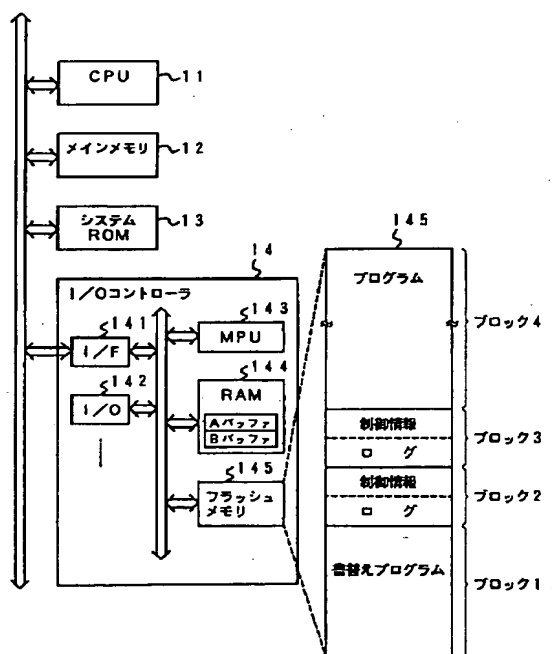
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 フラッシュEEPROMを用いたデータ処理装置

## (57) 【要約】

【目的】 従来のROMおよびEEPROM機能をフラッシュEEPROMのみによって実現し、部品点数を削減を図る。

【構成】 フラッシュEEPROM 145はブロック1〜ブロック4の4つのブロックに分割されており、書き換えプログラム、制御情報、システムプログラムがそこに格納される。制御情報はブロック2とブロック3に2重化されて格納されている。このため、一方の消去ブロックの更新中に電源遮断や他の何らかのエラーなどが発生しても、他方の消去ブロックの制御情報は正常に保持される。また、有効ブロックの制御情報が読み出されるため、正常な制御情報を利用する事ができる。したがって、制御情報の破壊を防止でき、従来のROMおよびEEPROM機能をフラッシュEEPROMのみによって実現することができる。



1.

## 【特許請求の範囲】

【請求項1】 ブロック単位でデータ消去可能なフラッシュEEPROMを有し、そのフラッシュEEPROMのプログラムを実行するデータ処理装置において、前記データ処理装置によって実行される機能を特定するための各種制御情報を2重化し、前記フラッシュEEPROMの第1および第2の消去ブロックに同一の制御情報を書き込む手段と、  
前記第1および第2の消去ブロックのどちらの制御情報を有効情報として使用するかを判断し、有効ブロックの制御情報を読み出す手段とを具備することを特徴とするデータ処理装置。

【請求項2】 前記2重化された各制御情報には誤り検出符号が付加されており、  
前記読み出し手段は、前記第1および第2の消去ブロックの内容が一致するか否かを検出し、一致する時には第1消去ブロックを有効ブロックとして決定し、不一致の時には各制御情報の誤り検出符号によって有効ブロックを決定することを特徴とする請求項1記載のデータ処理装置。

【請求項3】 前記書き込み手段は、  
制御情報に書き換え回数情報を付加する手段と、  
前記第1消去ブロックの消去および制御情報の書き込みによって前記第1消去ブロックの内容を更新する手段と、  
前記第1消去ブロックの更新後、前記第2消去ブロックの消去および制御情報の書き込みによって前記第2消去ブロックの内容を更新する手段とを具備することを特徴とする請求項1記載のデータ処理装置。

【請求項4】 前記読み出し手段は、前記第1および第2の消去ブロックの内容が一致するか否かを検出し、一致する時には第1消去ブロックを有効ブロックとして決定し、不一致の時には書き換え回数情報の値が大きいブロックを有効ブロックとして決定することを特徴とする請求項3記載のデータ処理装置。

【請求項5】 前記第1および第2の消去ブロックの各々は、前記制御情報が格納される制御情報格納部と、前記データ処理装置の動作履歴を示すログデータが格納されるログデータ格納部とに分割されており、  
前記第1および第2の消去ブロックのログデータ格納部をリング状に連なったエリアとして管理し、ログデータが採取される度にそのログデータを前記エリアに追記方式で書き込むログデータ書き込み手段をさらに具備することを特徴とする請求項1記載のデータ処理装置。

【請求項6】 前記ログデータ書き込み手段は、書き込み対象ブロックの前記ログデータ格納部の空きエリアサイズを定期的に検出し、空きエリアが所定サイズ以下になった時に他ブロックのログデータ格納部の内容を消去する手段を含むことを特徴とする請求項1記載のデータ処理装置。

2

【請求項7】 前記フラッシュEEPROMは、前記データ処理装置が実行するプログラムが格納されている第3の消去ブロックと、前記フラッシュEEPROMの内容を書き替えるための書き替えプログラムが格納され、書き替えが禁止されている第4の消去ブロックとをさらに有し、

前記書き替えプログラムを実行して前記第3消去ブロックのプログラムを書き替えるプログラム書き替え手段であって、前記第3消去ブロックのプログラム内容の不正を検出した時、または前記第3消去ブロックのプログラムの書き替えが指示された時にプログラム書き替えを実行するプログラム書き替え手段をさらに具備することを特徴とする請求項1記載のデータ処理装置。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、例えばマイコン組込み型のI/Oコントローラなどのデータ処理装置に関し、特にフラッシュEEPROMを有し、そのフラッシュEEPROMのプログラムを実行するデータ処理装置に関する。

【0002】

【従来の技術】一般に、コンピュータシステムにおいては種々のI/Oコントローラが設けられている。これらI/Oコントローラのほとんどは、マイコン組込み型のインテリジェントなデバイスによって構成されており、マイクロプロセッサ、ROM、RAM、およびEEPROMなどを内蔵している。

【0003】I/Oコントローラは、ROMに格納されたプログラムを実行することによって動作する。RAMはマイクロプロセッサの作業エリアとして使用され、またEEPROMはI/Oコントローラの機能を特定するパラメータなどの制御情報の格納に利用される。EEPROMに格納された制御情報はホストシステムからロードされるデータによって変更され、これによって例えばシステム起動時などにおけるI/Oコントローラの動作を制御する事ができる。

【0004】このように、従来では、プログラムと制御情報をそれぞれその特性に合わせて物理的に異なるメモリ上に格納していた。このため、部品点数は増加するものの、制御情報やプログラムの信頼性は比較的高く維持することができる。

【0005】しかし、その反面、ROMプログラムをバージョンアップする場合にはROMを交換する必要があるなどの不具合もある。そこで、最近では、ROMの代わりにフラッシュEEPROMが用いられるようになっていく。フラッシュEEPROMはブロック単位で書き換え可能であり、これを使用すればオンボード上でプログラムを容易に書き替える事ができる。

【0006】また、フラッシュEEPROMに制御情報も格納すれば、従来使用していたEEPROMが不用と

10

20

30

40

50

なり、部品点数の削減も可能となる。しかしながら、フラッシュEEPROMはブロック単位でデータ消去が行われるので、書き替え途中で電源遮断や他の何らかのエラーなどが発生すると全ての制御情報が破壊されてしまうという危険がある。このため、現在では、フラッシュEEPROMを使用した場合の信頼性の向上が要求されている。

【0007】

【発明が解決しようとする課題】従来では、ROMの代りにフラッシュEEPROMを用いることでプログラムの書き替え可能な構成にする事はできるが、信頼性の問題からEEPROMを別個に設ける必要があり、十分に部品点数を削減することはできなかった。

【0008】この発明はこのような点に鑑みてなされたもので、フラッシュEEPROMの書き替え途中で電源遮断や他の何らかのエラーなどが発生しても制御情報の破壊を防止できるようにし、従来のROMおよびEEPROM機能をフラッシュEEPROMのみによって実現することができるデータ処理装置を提供することを目的とする。

【0009】

【課題を解決するための手段】この発明は、ブロック単位でデータ消去可能なフラッシュEEPROMを有し、そのフラッシュEEPROMのプログラムを実行するデータ処理装置において、前記データ処理装置によって実行される機能を特定するための各種制御情報を2重化し、前記フラッシュEEPROMの第1および第2の消去ブロックに同一の制御情報を書き込む手段と、前記第1および第2の消去ブロックのどちらの制御情報を有効情報として使用するかを判断し、有効ブロックの制御情報を読み出す手段とを具備することを特徴とする。

【0010】このデータ処理装置においては、フラッシュEEPROMの2つの消去ブロックに制御情報が2重化されて格納されている。このため、一方の消去ブロックの更新中に電源遮断や他の何らかのエラーなどが発生しても、他方の消去ブロックの制御情報は正常に保持される。また、有効ブロックの制御情報が読み出されるため、正常な制御情報を利用する事ができる。したがって、制御情報の破壊を防止でき、従来のROMおよびEEPROM機能をフラッシュEEPROMのみによって

【0011】

【実施例】以下、図面を参照してこの発明の実施例を説明する。図1にはこの発明の一実施例に係るコンピュータシステム全体の構成が示されている。このコンピュータシステムは、CPU11、メインメモリ12、システムROM13、およびI/Oコントローラ14を備えている。I/Oコントローラ14は、このシステムに設けられるディスプレイモニタ、ハードディスク装置、RS232Cデバイス、フロッピーディスク装置などの各種

I/Oを制御するためのマイコン組み込み型の装置であり、図示のように、ホストシステムインタフェース141、各種I/Oに結合される複数のI/Oインタフェース142、マイクロプロセッサ143、RAM144、およびフラッシュEEPROM145を備えている。

【0012】RAM144は、マイクロプロセッサ143の作業領域として使用されるものであり、ここには2つのバッファAとバッファBの2つのバッファが確保されている。これらバッファは、フラッシュEEPROM145のデータ更新などのために利用される。

【0013】フラッシュEEPROM145は、I/Oコントローラ14のROMプログラムおよび制御情報格納用のEEPROMの機能を実現するために設けられたものであり、例えば128Kバイトの容量を持つチップによって実現されている。

【0014】このフラッシュEEPROM14はブロック1〜ブロック4の4個の消去ブロックに分割されており、ブロック1は8Kバイト、ブロック2は4Kバイト、ブロック3は4Kバイト、ブロック4は112Kバイトの容量を有している。

【0015】ブロック1には、フラッシュEEPROM14の書き替えプログラムが格納されている。ブロック2およびブロック3は、例えばシステム起動時に実行されるI/Oコントローラ14内の各デバイスの初期化処理に使用されるパラメータや、フロッピーディスク装置やハードディスク装置などのディスク装置の起動順位などを示すパラメータなど、マイクロプロセッサ143の機能を特定するための制御情報の格納に利用される。この場合、ブロック2およびブロック3には、同一の制御情報が2重化されて格納されている。

【0016】また、ブロック2およびブロック3の各々は、制御情報格納エリアとログ情報格納エリアに分割されている。ログ情報格納エリアには、このシステムの動作状況を管理するために必要な動作履歴を示すログデータ（電源ON/OFF時刻情報、システム障害情報など）が格納される。この場合、ブロック2およびブロック3のログ情報格納エリアは図2に示されているようにリング状に連なった1つのエリアとして管理され、ログデータが採取される度にそのログデータが追記方式で書き込まれる。

【0017】すなわち、ログ情報格納エリアについては2重化はなされておらず、ブロック2に設けられた512個のログエントリとブロック3に設けられた512個のログエントリとから構成される合計1024個のログエントリに、ログデータが順次格納される。各ログエントリは、例えば8バイトのサイズを有している。

【0018】また、ブロック2およびブロック3それぞれの制御情報格納エリアは、図3に示されているように、チェックサムデータ格納領域、チェックサム反転データ格納領域、書き替え回数データ格納領域、および制

御情報格納領域に分割されている。

【0019】フラッシュEEPROM14のブロック1〜ブロック4はそれぞれオンボード上で書き替え可能であるが、ブロック1についてはその書き替えが外部制御回路などによって禁止されている。

【0020】次に、図4乃至図11のフローチャートを参照して、フラッシュEEPROM145を利用して実行されるI/Oコントローラ14の動作を説明する。図4は、フラッシュEEPROM145のブロック1に納められた書き替えプログラムの動作を示すフローチャートである。

【0021】書き替えプログラムは、ブロック4のプログラムの更新（機能向上やバグ修正）がオペレータによって要求された場合や、プログラムが破壊された場合などに実行される。

【0022】プロセッサ143は、システム起動時に書き替えプログラムを実行し、まず、ブロック4のプログラム更新要求の有無を判定する（ステップS11）。この判定は、例えばオペレータによって所定のキー入力操作、またはスイッチ操作がなされたか否かを検出することによって行われる。プログラム更新要求がない場合は、プロセッサ143は、ブロック4に格納されているチェックサムを用いてブロック4の内容が正当であるか否かを判断する（ステップS12）。

【0023】プログラム更新要求が発行された場合、またはブロック4の内容のエラーが検出された場合には、プロセッサ143はフロッピーディスク装置やハードディスク装置などから新たなプログラムをロードし、そのプログラムのレビジョンを画面表示して正しいプログラムであるか否かをユーザに問い合わせる（ステップS13、S14）。

【0024】正しいことが確認されると、プロセッサ143は、ブロック4に格納されている書き替えプログラム内の書き替えルーチンをRAM144上にコピーし（ステップS15）、そのRAM144上で書き替えルーチンを実行する（ステップS16）。これにより、ブロック4の消去、ブロック4への新プログラムの書き込みが行われ、その後、書き替えプログラムからブロック4の新プログラムへ制御が移される。

【0025】このように、ブロック1の書き替えプログラムは、ブロック4の書き替え指示が有った時、又はフラッシュEEPROM145の内容の正当性のチェックを行ない、不正であれば、フラッシュEEPROM145の内容の書き替えを行なう全ての機能（1. 新プログラムの読み込み、2. 読み込みプログラムのバージョン確認、3. フラッシュメモリの書きき替え、4. 書き込み後の再スタート）を持っている。

【0026】図5は、2重化された制御情報の更新を行なうフローチャートである。制御情報を更新する場合に

が転送される。CPU11による転送制御は、制御情報がRAM144のバッファAに格納された時点で完了し、バッファAからフラッシュEEPROM145へのデータ書き込みはI/Oコントローラ14内部の処理として実行される。

【0027】すなわち、プロセッサ143は、バッファAにあらかじめ制御情報格納エリアの内容を読み出しておき、そのバッファAにCPU11から制御情報が転送されると、まず、バッファA上の書き替え回数データの値を+1カウントアップする（ステップS21）。次いで、プロセッサ143は、バッファAの内容（転送された制御情報、および書き替え回数データを含む）をバッファBにコピーした後、チェックサムを生成して図3に示した構造を持つ制御情報ブロックを作る（ステップS23）。その後、プロセッサ143は、ブロック2の更新を行ってから、ブロック3の更新を行う（ステップS24、S25）。

【0028】これら各ブロックの更新処理では、まず、ログ格納部の内容がバッファBに読み出されて、ログデータが制御情報に付加される（ステップS31）。次いで、ブロック消去、バッファBからブロックへのデータ書き込みが行われる（ステップS32〜S34）。

【0029】このように、制御情報の更新は、要求時にバッファAを更新し、書き込み時に、バッファA上の書き替え回数を更新した後、バッファAをバッファBにコピーし、チェックサムを作成し、ブロック2を更新後、ブロック3を更新するという手順で行われる。また、ブロック2、ブロック3にはログデータが有るため更新時はこの値をフラッシュメモリから読み出し、フラッシュ後再度書き込む。フラッシュメモリ更新時、フラッシュした時と同じ値のデータは書き込まれない。例えば、データ消去によって各ビットの記憶内容が“1”に設定される場合には、データ“0”の書き込みだけが行われる。

【0030】このような更新処理により、ブロック2およびブロック3には同一の制御情報が2重化されて格納される。また、一方のブロック更新終了後に、もう一方を更新するという手続きで消去および書き込みが行われるので、制御データ更新中に障害が発生しても片方の制御情報は消失されずに保持される。

【0031】図6は2重化した制御情報の内、どちらを有効情報として利用するかを判定するためのフローチャートであり、この判定処理はシステム立上時に実行される。すなわち、プロセッサ143は、まず、ブロック2とブロック3の制御情報を比較して、それら等しいか否かを検出する（ステップS41）。等しい場合にはブロック2とブロック3のどちらの制御情報も正当なものであるため、いずれか一方、例えばブロック2が有効ブロックであると判断される（ステップS42）。

【0032】ブロック2とブロック3の制御情報が不-

10

20

30

40

50

致の場合には、プロセッサ143は、ブロック3のチェックサムデータを用いてブロック3の内容の正当性を調べる(ステップS43)。ブロック3の内容が正しくない場合(チェックサムチェックエラー)には、ブロック2が有効ブロックであると判断される(ステップS44)。

【0033】ブロック3の内容が正しい場合(チェックサムチェック成功)には、プロセッサ143は、ブロック2のチェックサムデータを用いてブロック2の内容の正当性を調べる(ステップS45)。ブロック2の内容が正しくない場合には、ブロック3が有効ブロックであると判断される(ステップS46)。

【0034】ブロック2とブロック3の双方とも内容が正しい場合には、プロセッサ143は、更新回数の多い方のブロックを有効とする(ステップS47)。このようにして有効と判定されたブロックの制御情報は、立上げ時にバッファAに読み出され、以後、バッファAの値を利用した動作制御が行われる(ステップS48)。

【0035】図7は、ブロック2、ブロック3に設けたログエリアの書き込みポイントを見つけるフローチャートである。システム立ち上げ時、プロセッサ143は、ブロック2のログエリアの先頭(ログエントリ1)にログデータが格納されているか否かを検出する(ステップS51)。この検出は、ログエントリ1の内容がオール“1”であるか否かを調べることによって行われる。

【0036】ブロック2のログエリアの先頭にログデータが無い場合には、プロセッサ143は、ブロック3のログ未書き込みエリアを探し、未書き込みエリアが存在する場合にはその先頭エントリを書き込みポイントとして決定する(ステップS52)。もしブロック3にログ未書き込みエリアが存在しない場合には、プロセッサ143は、ブロック2のログエリアの先頭エントリを書き込みポイントとして決定する(ステップS53)。

【0037】また、ブロック2のログエリアの先頭にログデータが存在する場合には、プロセッサ143は、ブロック3のログ未書き込みエリアを探し、未書き込みエリアがある場合にはその先頭エントリを書き込みポイントとして決定する(ステップS54)。ブロック2に未書き込みエリアが存在しない場合には、プロセッサ143は、ブロック3のログ未書き込みエリアを探し、未書き込みエリアがある場合にはその先頭エントリを書き込みポイントとして決定する(ステップS55)。ブロック3にログ未書き込みエリアが所定数以上存在しない場合には、プロセッサ143は、ブロック2のログエリアを消去し(ステップS56)、ブロック2のログエリアの先頭エントリを書き込みポイントとして決定する(ステップS53)。

【0038】このように、ブロック2およびブロック3のログエリアはリング状に連続した1つのエリアとして使用され、ログ書き込みポイント側のブロックの未書き

込みログエリアが少なくなると、反対側のログエリアがフラッシュされる。未書き込みログエリアの残り数のチェックは通常動作時においても定期的に行ない、未書き込みログエリアが亡くなる前にフラッシュによってログエリアを確保しておくことが好ましい。また、ログ未書き込みエリアが残っていない場合には、ログを採取しないなどの運用を行うこともできる。

【0039】図8は、ログエリアを確保するために行われるログフラッシュ動作を示すフローチャートである。ここでは、ブロック3のログエリアに空きが少なくなり、ブロック2のログエリアを消去する場合を例示して説明する。

【0040】プロセッサ143は、まず、ブロック2の内容(制御情報エリアおよびログエリア)をバッファBに読み出し、ログエリアの内容だけをオール“1”にする(ステップS61)。次いで、プロセッサ143は、ブロック2の消去を行う(ステップS62)。この後、プロセッサ143は、バッファBの内容をブロック2に書き込む(ステップS63)。この場合、バッファBのログエリアの内容は“1”であるため、制御情報エリアの内容だけがブロック2に書き込まれ、バッファBのログエリアの内容については書き込みは行われない。このようにして、ブロック2のログエリアだけが消去される。

【0041】図9は、採取されたログデータをログエリアに書き込む動作を示すフローチャートである。プロセッサ143は、CPU11からログデータの書き込み要求が発行されると、まず、現在書き込み対象となっているブロック内に未書き込みログエントリが所定数、例えば2個以上残っているか否かを調べる(ステップS71)。2個以上残っていれば、プロセッサ143は、採取したログデータを書き込みポイントで指定される位置に書き込み、その後、書き込みポイントの値を更新しておく(ステップS72、S73)。

【0042】次に、プロセッサ143は、更新された書き込みポイントの値がそのブロックの最終エントリを示しているか否かを調べ、最終エントリを示しているならば、書き込みポイントの値を他方のブロックの先頭エントリに変更する(ステップS74、S75)。

【0043】一方、ステップ71にて未書き込みログエントリの残り数が1個だけあることが検出されると、他方のブロックの先頭ログエントリが未書き込みであるか否かを調べる(ステップS76)。未書き込みの場合には、その先頭ログエントリを書き込みポイントとして書き込み対象ブロックを変更し、そして、ステップS72～S75の処理を行う。先頭ログエントリが書き込み済みの場合には、ログ書き込みを行わずに処理を終了する。

【0044】図10および図11は通常動作時におけるフラッシュEEPROM145に対するアクセス動作(制御データ更新、ログ書き込み要求、ログ未書き込み

エリアチェック、制御データ読み出し、プログラム更新)を示すフローチャートである。

【0045】この実施例では、フラッシュEEPROM145を4つのブロックに分けて管理し、データ内容毎に異なる機能を提供しているため、フラッシュEEPROM145に対する複数の要求が同時に発生する。

【0046】ログ書き込み要求は即時フラッシュメモリに書き込み、他の要求は、フラッシュメモリアクセスタスクで実現し、同時に発生した要求を順番に処理する。タスク動作中に発生したログ書き込みは、タスク動作終了時まで待たせ、タスクで書き込みを行なう。又、制御情報の更新要求はタスク動作中複数発生しても一度の更新で行なう様、タスク動作中は更新フラグを立てておき、タスク動作後に更新フラグがあれば、制御情報の更新を行なう。

【0047】まず、図10(A)を参照して、制御データ更新要求が発行された場合の動作を説明する。プロセッサ143は、CPU11から制御情報更新要求を受け取ると、バッファAに読み出されている制御情報をそのバッファA上で更新し(ステップS81)、その後、タスク動作中であるか否かを調べる(ステップS82)。タスク動作中であれば更新フラグをオンにして処理を終了し、タスク動作中でなければタスクを起動する(ステップS83、S84)。

【0048】また、ログ書き込み要求が発行されると、図10(B)に示されているように、プロセッサ143は、タスク動作中か否かを調べ、タスク動作中はログを書き込まずログデータを保存しておく(ステップS91～S93)。

【0049】ログ未書き込みエリアのチェック処理は定期的に呼び出され、この場合には、図10(C)に示されているように、プロセッサ143は、ブロックの未書き込みエリアが少なくなっていることを検出すると、他方のログエリアをあらかじめフラッシュする(ステップS101～S103)。

【0050】制御情報の読み出し要求については、図10(D)に示されているように、システム起動時に有効ブロックからバッファAに読み出した内容が使用される(ステップS111)。

【0051】プログラム変更要求が発行されると、図10(E)に示されているように、タスクの起動が行われる(ステップS121)。次に、図11を参照して、タスクによる動作手順を説明する。

【0052】プロセッサ143は、タスクの初期化時において、有効ブロックの制御情報をバッファAに読み出しておき(ステップS201)、以降、起動要求を判断して(ステップS202、S203)、その起動要求に対応する処理(プログラム更新、制御データ更新フラグOFFおよび制御データ更新、保存ログまたは採取ログの書き込み、ログフラッシュなどを行う(ステップS2

04～S211)。

【0053】以上のように、この実施例においては、フラッシュEEPROM145を4つのブロックに分け、装置上でフラッシュEEPROM145の更新を可能にし、また外部から更新データを読み込み機能を有する事で、従来ROM交換等に対応していた、機能向上や、バグの修正を装置自身で実現できる。また、制御情報を2重化してフラッシュEEPROM145内に置く事で、制御データ更新中に障害が発生しても片方の制御情報が失われないので、信頼性の向上を実現できる。又、フラッシュメモリの制御情報ブロックの未使用エリアをログデータエリアとして使用しており、不安定動作の解析、特に電源切断時の誤動作解析に有効に利用する事ができる。さらに、ログデータの書き込みにおいては2つのログエリアをリング状の連続したエリアとして管理し、追記方式で書き込みを行うので、採取したログを空きエントリに即時に書き込む事ができる。

【0054】

【発明の効果】以上説明したように、この発明によれば、フラッシュEEPROMの2つの消去ブロックに制御情報が2重化されて格納されているため、一方の消去ブロックの更新中に電源遮断や他の何らかのエラーなどが発生しても、他方の消去ブロックの制御情報は正常に保持される。また、有効ブロックの制御情報が読み出されるため、正常な制御情報を利用する事ができる。したがって、制御情報の破壊を防止でき、従来のROMおよびEEPROM機能をフラッシュEEPROMのみによって実現することができる。

【図面の簡単な説明】

【図1】この発明の一実施例に係わるコンピュータシステムの全体の構成を示すブロック図。

【図2】同実施例のコンピュータシステムに設けられたフラッシュメモリのブロック利用形態を説明するための図。

【図3】同実施例のコンピュータシステムに設けられたフラッシュメモリに2重化されて書き込まれる制御情報のデータ構造を示す図。

【図4】同実施例のフラッシュメモリに格納された書き替えプログラムの動作を示すフローチャート。

【図5】同実施例のフラッシュメモリに2重化されて格納されている制御情報の更新動作を示すフローチャート。

【図6】同実施例の2重化された制御情報の内、どちらかを有効情報として利用するかを判定するための動作を示すフローチャート。

【図7】同実施例のフラッシュメモリのログ書き込みポイントを見つける動作を示すフローチャート。

【図8】同実施例のフラッシュメモリにログエリアを確保するために行われるログフラッシュ動作を示すフローチャート。



11

【図9】同実施例のフラッシュメモリにログデータを書き込む動作を示すフローチャート。

【図10】同実施例のフラッシュメモリの各種アクセス要求に対する動作を説明するフローチャート。

【図11】同実施例のフラッシュメモリに対する各種アクセス要求に対して実行されるタスクの動作を説明するフローチャート。

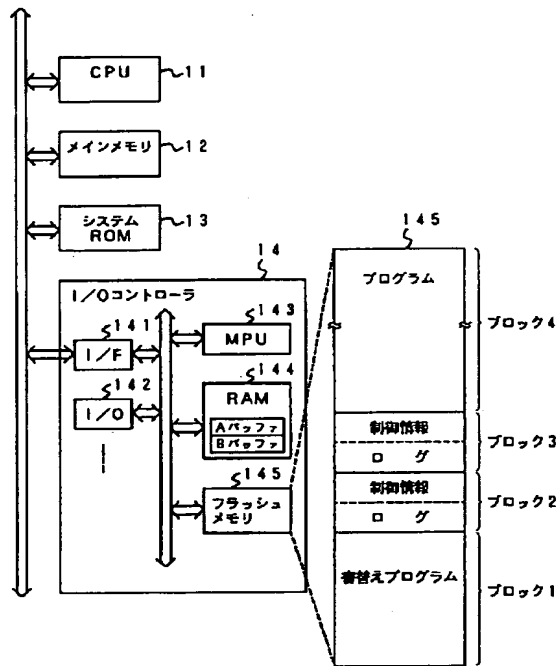
\*

12

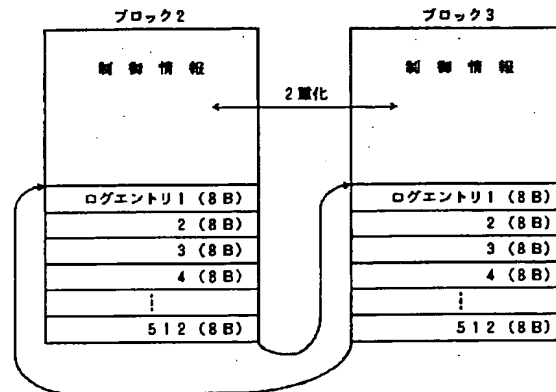
\*【符号の説明】

11…CPU、12…メインメモリ、13…システムROM、14…I/Oコントローラ、141…ホストインタフェース、142…I/Oインタフェース、143…マイクロプロセッサ、144…RAM、145…フラッシュEEPROM。

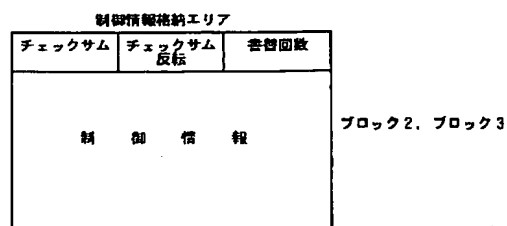
【図1】



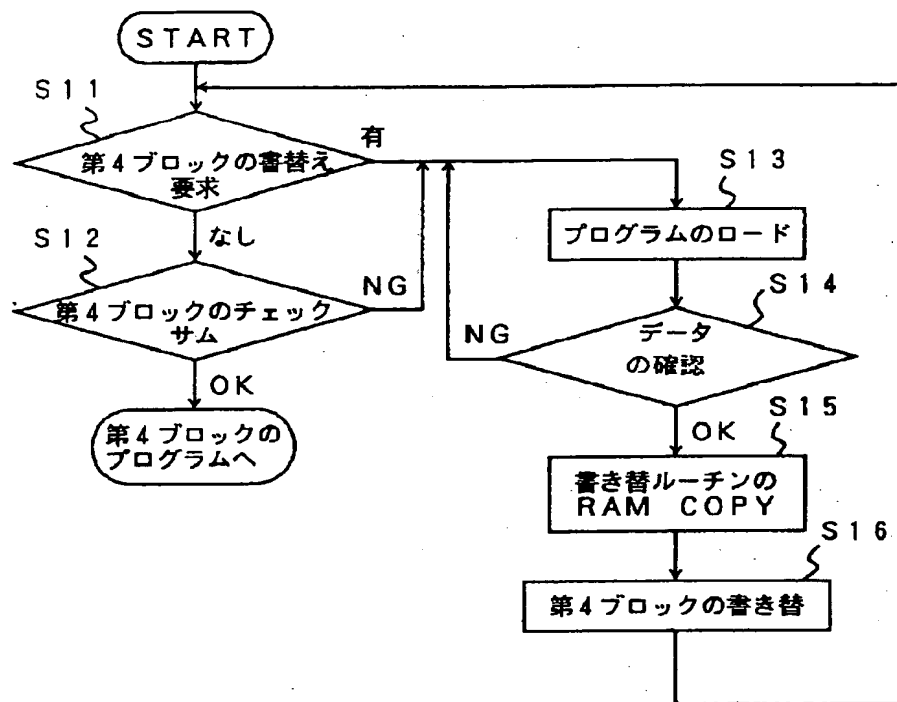
【図2】



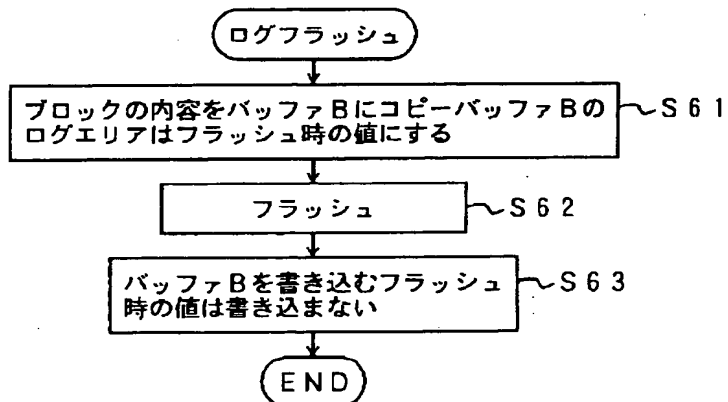
【図3】



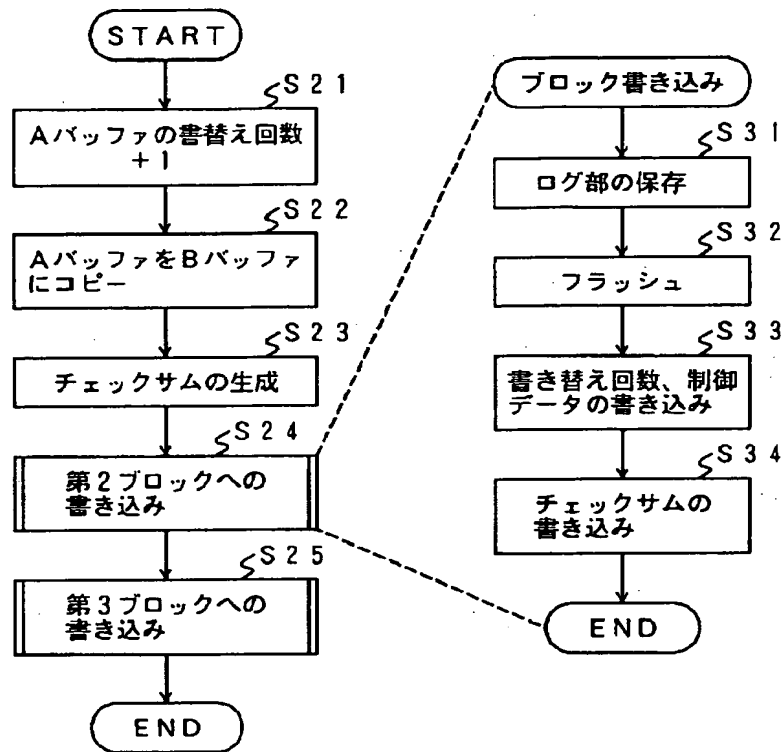
【図4】



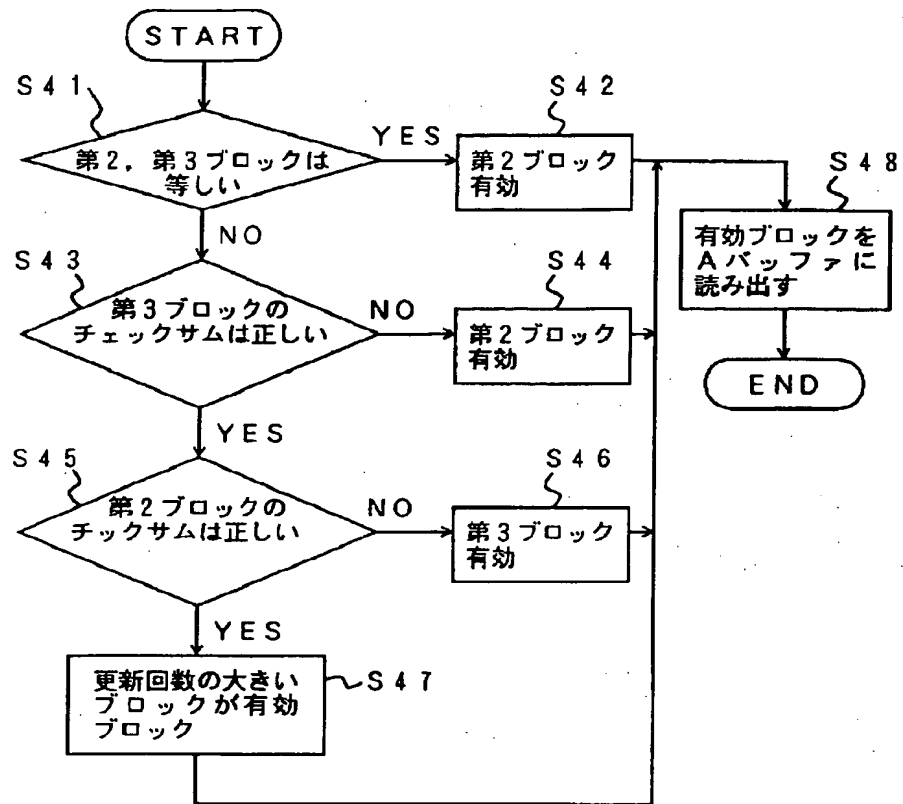
【図8】



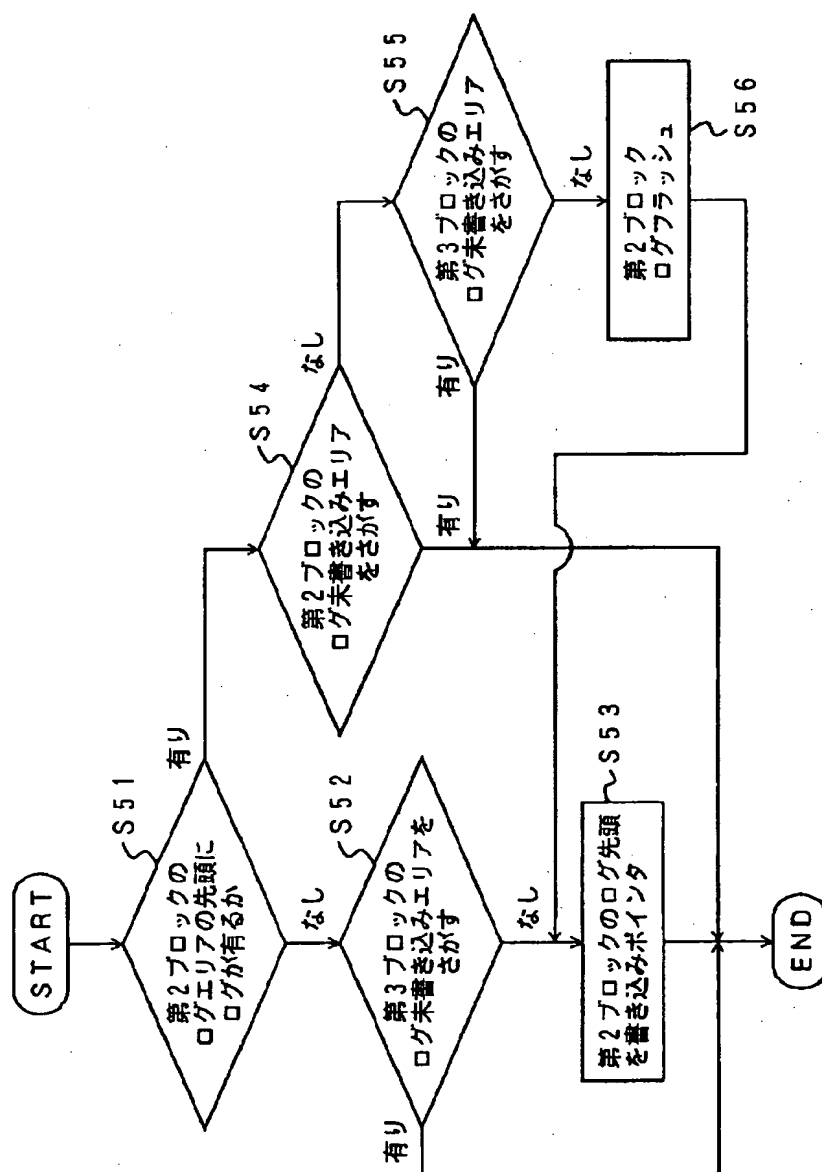
【図5】



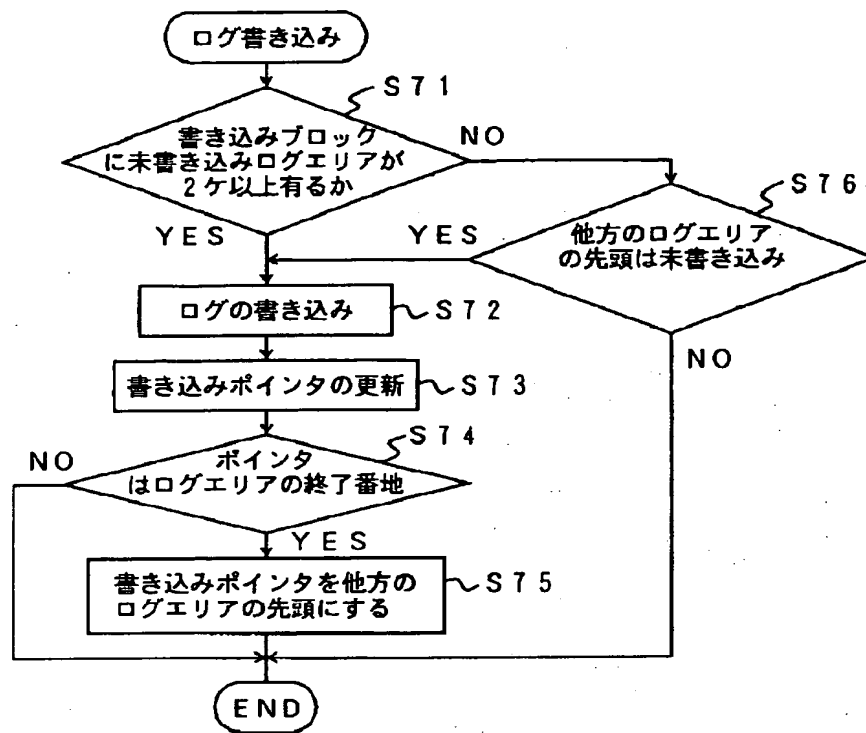
【図6】



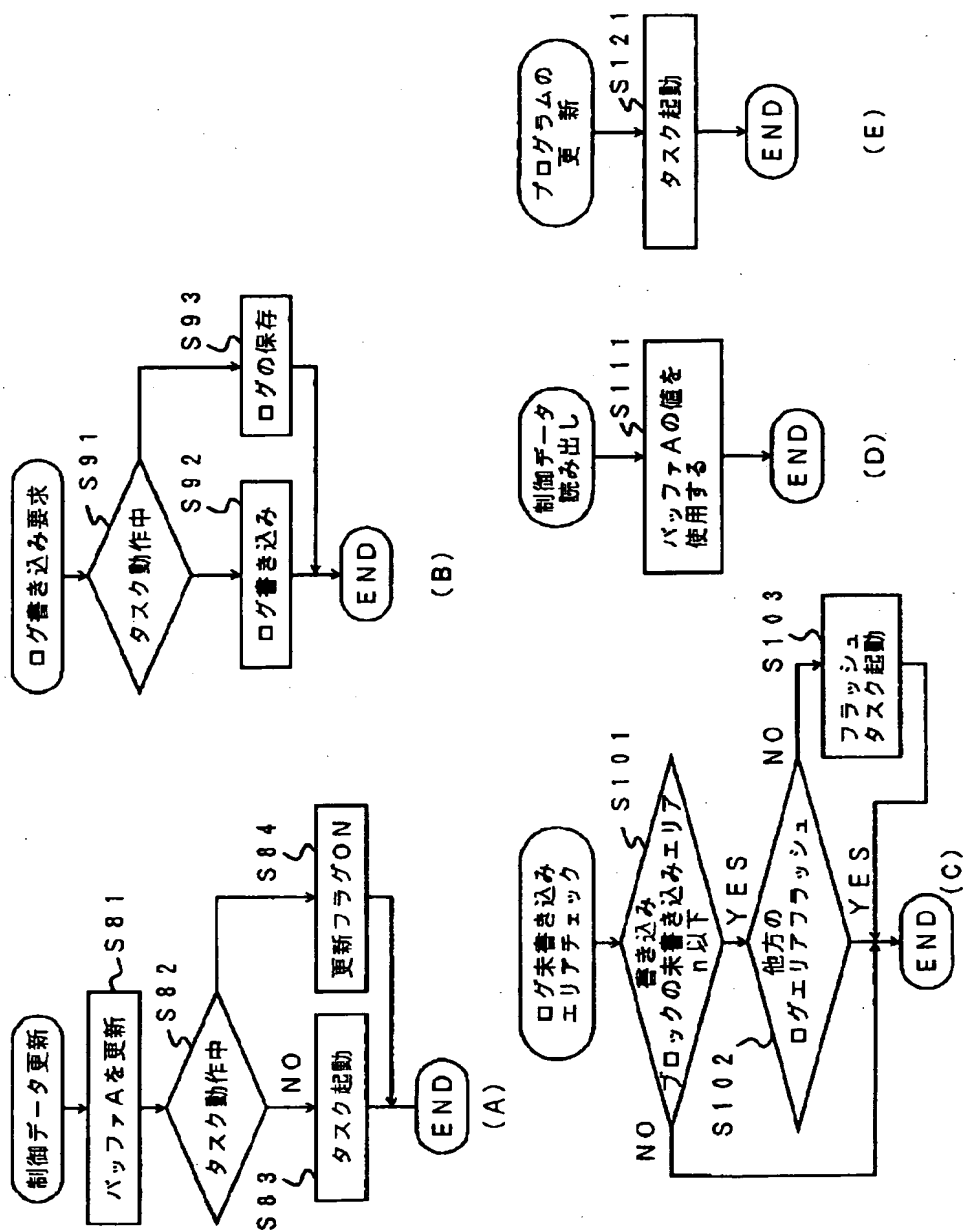
【図7】



【図9】



【図10】



【図11】

